Exercise 3

Symbolic Model Checking

1) Give the initial predicate and the transition predicate for the following Verilog module:

```
module simple(clock, flip1, flip2, var1, var2);
   input clock, flip1, flip2;
   output var1, var2;
   reg      var1, var2;

   initial begin
      var1 = 0;
      var2 = 0;
   end

   always @(posedge clock) begin
      if (flip1) begin
          var1 = ~var1;
      end
      if (flip2) begin
          var2 = ~var2;
      end
   end
endmodule
```

2) Consider the Boolean expression $(x_1 \wedge x_2 \wedge x_3) \vee (\neg x_2 \wedge x_4) \vee (\neg x_3 \wedge x_4)$

Choose a variable ordering for the variables x_1, x_2, x_3, x_4, and draw the resulting BDD. Can you reduce the size of the BDD by reordering the variables?

3) Let X be the set $\{x_0, x_1, y_0, y_1, out_0, out_1, carry\}$. Choose an appropriate ordering of the variables, and construct the BDD for the requirement that the output out_1out_0, together with the carry bit carry, is the sum of the inputs x_1x_0 and y_1y_0. Is your choice of ordering optimal?

4) Given a BDD B over a set X of variables, a variable x, given an algorithm to construct a BDD representing the function $\exists$x.B, so x is existentially quantified over B. What is the running time of your algorithm in terms of the number of vertices of the input B.

Bonus: Instead of pseudo-code, extend miniBDD http://www.cprover.org/miniBDD/ with a function with the signature: `BDD exists(unsigned) const;`
that when applied to BDD B with the variable id x returns a BDD that corresponds to $\exists$x.B.